In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

7/25/68

# CONTROL THEORY METHODS FOR THE SOLUTION OF MATHEMATICAL PROGRAMMING PROBLEMS

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

by

Massimo Actis Dato

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in the School of Industrial and Systems Engineering

Georgia Institute of Technology

August, 1971

CONTROL THEORY METHODS FOR THE SOLUTION

OF MATHEMATICAL PROGRAMMING PROBLEMS

Approved:

Chairman

Date approved by Chairman: 6 Aug 1971

## ACKNOWLEDGMENTS

My deepest appreciation and respect are expressed to Dr. L. G. Callahan, who has unfailingly provided sound advice, knowledgeable guidance and cheerful encouragement to me during the conduct of this research.

Special acknowledgment is also due Dr. J. J. Talavage, who as my Faculty Advisor and member of my thesis advisory committee, has provided the fundamental guidance to my studies in the School of Industrial Engineering.

The author also would acknowledge the help of Dr. C. M. Shetty, who with willing and able assistance contributed greatly in the formulation of this study.

Finally, I am indebted to Mrs. Betty Roper Sims for her skillful assistance in typing this manuscript.

# TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF ILLUSTRATIONS

# SUMMARY

This thesis is concerned with the development of new methods for the solution of mathematical programming problems.

The most important solution procedure in the linear case is Dantzig's Simplex Method, which has been designed for use on digital computers. Other procedures are due to Jackson (1957) and to Pyne (1956). They require the use of an analog computer, and they seem to have some advantages over the Simplex Method. In the first part of the thesis, these two procedures are described, and their performances evaluated. The velocity by which the solution to the problem can be obtained is considered, and it is shown how it can be considerably greater than Dantzig's Method, particularly if the problem is of large dimensions. It is also shown how nonlinear problems can be framed so that they can be solved applying a technique similar to the one used in the linear case. Sensitivity analysis can be performed quite easily, in a matter of seconds, and it does not require a rerun of the whole problem.

A drawback of those analog methods is the lack of precision. In the second part of this thesis it is tried to overcome this difficulty, either by improving the existing procedures, or by a new approach. The first method of attack leads to the use of a second order differential equation model, which has the additional advantage of increasing the solution speed; the second consists in reframing the mathematical model,

and leads to the simultaneous solution of the primal and dual problems. The two suggested improvements can be combined, adding in such a way to the advantages.

Applications of the methods to a simple problem are given.

# CHAPTER I

# INTRODUCTION

## Nature of the Problem

Mathematical programming deals with the problem of allocating limited resources among competing activities in optimal manner. This problem of allocation can arise whenever one must select the level of certain activities which must compete for certain scarce resources necessary to perform those activities. The great variety of situations to which mathematical programming can be applied is indeed remarkable. It ranges from the allocation of production facilities to products to the allocation of airplane fuel to bomber runs, from portfolio selection to the selection of shipping patterns, and so on almost to infinitum. Mathematical programming uses a mathematical model to describe the problem of concern. The word "programming" is essentially a synonym for planning. Thus mathematical programming involves the planning of activities in order to obtain an "optimal" result, i.e. a result which reaches the specified goals best (according to the model) among all feasible alternatives. A great deal of attention has been given recently to the computational aspects of mathematical programming, and in particular of linear programming, in which all the mathematical functions in the model are required to be linear functions. The best known computational method is the simplex method of George Dantzig, designed for use on digital computing machinery for the maximization of a linear function of variables subject to linear inequalities. In the simplex method, a large number of arc computations involved (about 5 $m^3$, where m is the number of restrictions, according to (1)) and entails roundoff problems.

It is felt that many engineers and the majority of people in other fields are not aware of the potentialities of electronic analog computers. Actually, analog computation is a very simple approach to problem solutions, even if the concepts at first may seem somewhat illusive, and solves in a reasonable time.

Moreover, a change in the given data usually requires a rerun of the whole problem.

## Purpose of the Research

Methods designed for use on analogue computers are by far less known, even if they can be especially useful in removing some of the difficulties found using the simplex method.

In many practical problems the data are known with so little accuracy (much less than that of the values of the electrical components representing them) that it is important to be able to explore the solutions of a whole class of problems with data near those of the given problems. Since analog computer methods allow these changes to be made and solutions to be found quickly and examined for "reasonableness," uncertainty regarding data should no longer present as great an obstacle as heretofore in the application of linear programming to realistic problems.

Many of the problems are characterized by complex interaction of forces and highly nonlinear relationships, by the low precision of input data and the system parameters of only two or three significant figures, and by the need for a factor of human reasoning during problem solution. Certain nonlinear programming problems can be phrased as linear problems with variable coefficients, so that analog methods can be useful. The operating time of the analog computer makes possible the inclusion of the operator's judgment and creative reasoning in the problem solution since he can often immediately see the results of his decision and how they affect the validity of the system model and/or the problem solution. It is the purpose of this thesis to point out some of the ways the modern electronic analog computer can be used to advantage in the solution of such problems, and to stimulate interest and the conception of ideas in this method of problem solution. First two methods suggested by literature will be described. Next the limits of their performances will be evaluated. Finally we will consider how they can be improved, either choosing a new approach, or by a refinement of the existing solution procedures. The methods will be judged from the point of view of:

a. Maximum size of a solvable problem.

b. Velocity of execution.

c. Errors.

All of them are equally useful in the nonlinear case, and if sensitivity analysis is required.

# CHAPTER II

# LITERATURE SURVEY

## Jackson's Method

This method (1) is a logical extension of the method of "steepest ascent." A digital computer would solve a programming problem in a step-by-step numerical calculation. An analog computer will solve such a problem in a dynamic fashion even though the problem itself is static. The advantage of solving a problem in a dynamic fashion is one of speed, since the solution point is calculated in a continuous manner rather than in a discontinuous point-by-point method, and the resulting calculation for even a large programming problem on an electronic analog computer takes only a matter of seconds.

Although the method is capable of solving non-linear optimization problems, it will be illustrated by a single type which can be stated in the following forms. It is required to maximize the objective function of n variables.

$$z = \sum_{k=1}^{n} c_k x_k \qquad (x_k \geq 0) \tag{2-1}$$

subject to:

$$\sum_{k=1}^{n} x_k a_{ik} \leq R_i \qquad i=1,2,\ldots,n \tag{2-2}$$

Equation (2-2) can be expressed as:

$$\sum_{k=1}^{n} x_k a_{ik} \leq 1 \qquad i=1,2,\ldots,m \tag{2-3}$$

where the c's and a's are constants. The problem is solved with (n+m+2p) amplifiers, where

n = number of variables.

m = number of restrictions.

p = number of negative quantities among the a's.

The variables $x_1, x_2, \ldots, x_m$ can be treated as the coordinates of a point in a Euclidean space of n dimensions. The point with these coordinates will be called the objective point. The objective function, equation (2-1), is continuous and single valued everywhere within this space and therefore can be used to define a gradient (grad) vector, grad z. If $i_1, i_2, \ldots, i_n$ are unit vectors in the directions of the coordinate axes, then

$$\text{grad } z = \sum_{k=1}^{n} \frac{\partial z}{\partial x_k} i_k = \sum_{k=1}^{n} c_k i_k$$

The vector grad z is everywhere constant, normal to the hyperplanes of equal z, and in the direction of steepest ascent.

The $j$*th* restriction of equation (2-3) represents a half space in the n-space bounded by the hyperplane

$$\sum_{k=1}^{n} x_k a_{jk} = 1$$

An edge separates the region of the space in which an inequality is satisfied from the region in which it is violated. The space in which all the given inequalities are satisfied is called the feasible region. The allowed region is ordinarily bounded in the sense that the objective function cannot increase without limit without entering the restricted region. The scalar $1/a_{jk}$ represents the intercept of the $j$*th* edge with the $K$*th* coordinate axis. Any of the inequalities, equation (2-3), is superfluous if its hyperplane lies wholly within the infeasible region. Define a vector $N_j$ normal to the $j$*th* edge and directed toward the feasible region. There are exactly m such vectors and each is associated with a particular restriction and are given by

$$N_j = -(a_{j1}i_1+a_{j2}i_2+\ldots+a_{jn}i_n) \qquad (2\text{-}4)$$

Next, define m quantities $d_1,\ldots,d_m$, one for each inequality, such that

$$\begin{aligned} d_j &= 0 \quad \text{when} \quad \sum_{k=1}^{n} a_{jk}x_k \le 1 \\ d_j &= 1 \quad \text{when} \quad \sum_{k=1}^{n} a_{jk}x_k > 1 \end{aligned} \qquad (2\text{-}5)$$

Define the vector f as follows

$$f = K \text{ grad } z + H \sum_{j=1}^{m} d_j N_j \left( \sum_{k=1}^{n} x_k a_{jk} - 1 \right) \qquad (2\text{-}6)$$

where H and K are constants.

In terms of components, f may be written

$$f = f_1 i_1 + f_2 i_2 + \ldots + f_n i_n \tag{2-7}$$

from equations (2-3), (2-5), and (2-7), the *Kth* component is

$$f_k = Kc_k - H \sum_{j=1}^{m} a_{jk} d_j \left( \sum_{k=1}^{n} x_k a_{jk} - 1 \right) \tag{2-8}$$

Let the position of the objective point be described by the vector

$$r = x_1 i_1 + x_2 i_2 + \ldots + x_n i_n \tag{2-9}$$

The velocity of the objective point in the n-dimensional space may therefore be defined as the vector

$$v = \frac{dr}{dt} \tag{2-10}$$

where t is the time.

The equation which describes Jackson's method can now be written

$$\frac{dr}{dt} = f \tag{2-11}$$

Equation (2-11) describes the motion of the objective point as the solution is approached. The solution is to be the value which r finally attains.

The velocity of the objective point $P(x_1,x_2,\ldots,x_m)$ lies along the gradient of the objective function while in the n-dimensional solution space; hence the point moves in the direction of steepest ascent (descent). (The signs in equation (2-8) are reversed if it is desired to minimize z.) At points beyond the *jth* edge the velocity vector has a component along $N_j$, and for some finite value of $\left(\sum_{k=1}^{n} a_{jk}x_k - 1\right)$ this will just cancel out the component contained in z in the direction normal to the edge.

The velocity vector will then lie parallel to the *jth* hypersurface for finite positive values of $\left(\sum_{k=1}^{n} a_{jk}x_k - 1\right)$. At points in the neighborhood of relative maxima (minima) of z, dr/dt will vanish.

The above equations may be instrumented quite easily on an analog computer. In addition to summers, needed to solve the inequalities, and of integrators, needed to integrate the rates of change of the variables, some type of decision element must be used to generate the $d_k$. The most convenient way to perform this decision function is through the use of diode limiting circuits.

The resulting general computer schematic is given in Figure 1, where

$$e_j = -d_j\left(\sum_{k=1}^{n} a_{jk}x_k - 1\right)$$

The quantities of $x_j$ are integrated with respect to time from the starting position determined by the set of initial conditions $x_1(0)$, $x_2(0),\ldots,x_n(0)$. The quantity K is an arbitrary, large, real number. The effect of this feedback is to cancel the component in each $\dot{x}_i$ which

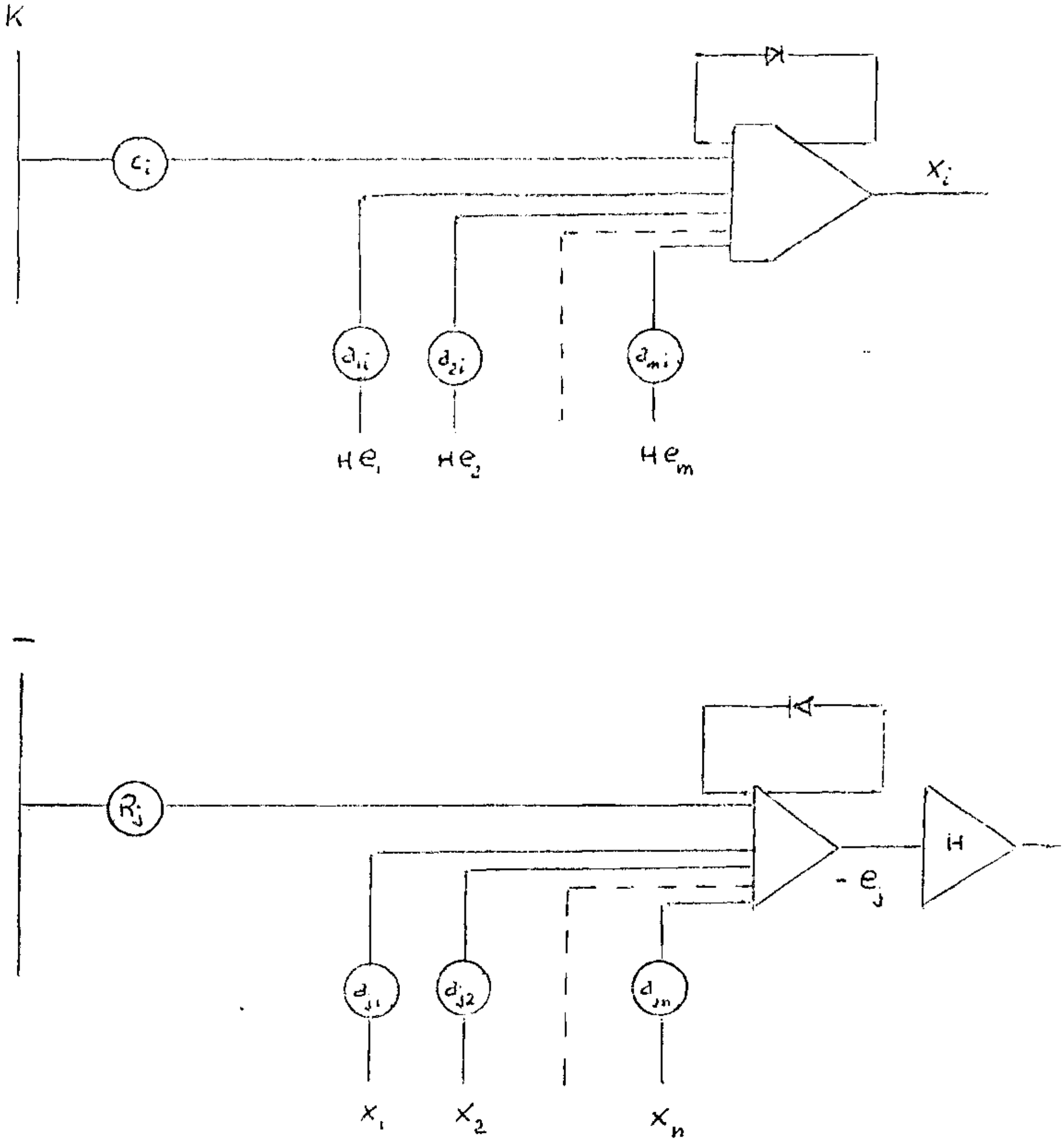


Figure 1. General Computer Schematic for Jackson's Method

is normal to that particular boundary hypersurface. Therefore, the point P is constrained to move in the direction of increasing $z(x)$, but parallel to the boundary surfaces and is in the infeasible region close to the boundary. The velocity of the point P is thus altered each time a boundary or restriction is reached. The velocity vector will always have a component along grad z until the maximum value of z is reached. When the maximum value of z is reached, the last component of the velocity in the direction grad z will have been cancelled and the point P will come to a rest. The maximum found in this way is determined uniquely by the defining equations and is independent of the starting or initial point.

## Pyne's Method

This method (2) solves the problem with $(n+2m+2p)$ amplifiers where

n = number of variables.

m = number of restrictions.

p = number of negative quantities among the restrictions' coefficients.

The vectors

$$\text{grad } z = \sum_{k=1}^{n} \frac{\partial z}{\partial x_k} i_k = \sum_{k=1}^{n} c_k i_k$$

$$N_j = -(a_{j1}i_1+a_{j2}i_2+\ldots+a_{jn}i_n)$$

and the quantities

$$d_j = 0 \quad \text{when} \quad \sum_{k=1}^{n} a_{jk}x_k \leq 1$$

$$d_j = 1 \quad \text{when} \quad \sum_{k=1}^{n} a_{jk}x_k > 1$$

are defined in the same way as in Jackson's method, but there is a difference in the definition of vector f:

$$f = K \text{ grad } z + H \sum_{j=1}^{n} d_j N_j \qquad (2\text{-}12)$$

In terms of components f may be written

$$f = f_1 i_1 + f_2 i_2 + \ldots + f_n i_n$$

From equations (2-3), (2-5) and (2-7), the K*th* component is

$$f_k = Kc_k - H \sum_{j=1}^{n} d_j a_{jk} \qquad (2\text{-}13)$$

Both the vector f and its K*th* component $f_k$ depend on the coordinates of the objective point because of the presence of the d's in equations (2-12) and (2-13).

Let the position of the objective point be described by the vector

$$r = x_1 i_1 + x_2 i_2 + \ldots x_n i_n$$

The velocity of the objective point in the n-dimensional space may therefore be defined as the vector

$$v = \frac{dr}{dt} = f \tag{2-14}$$

Equation (2-14) describes the motion of the objective point as the solution is approached. The solution is to be the value which r finally attains. From equation (2-12) the K*th* component of equation (2-14) may be written

$$\frac{dx_k}{dt} = Kc_k - H \sum_{j=1}^{m} d_j a_{jk} \tag{2-15}$$

The objective point moves through the allowed region with a velocity K grad z, until it reaches a restriction such as the j*th*. At this point, the motion is determined by two vectors, the gradient and the vector $N_j$, normal to the j*th* hyperplane. If $N_j$ is greater than the normal component of K grad z, the point will be ejected from the restricted region. The actual rebound is infinitesimal in magnitude as the objective point moves into the allowed region. The vector $d_jN_j$ then vanishes, and the gradient causes the motion to reverse until the point again enters the restricted region. In this way, the objective point moves along the restriction boundary in the direction of the projection of the vector gradient z on the j*th* hyperplane. To solve equation (2-15) on an analog computer, it is necessary to divide the computer into the following three types of systems:

1. A set of n summing integrators limited to allow positive outputs only.

2. A set of m summing amplifiers.

3. A set of m switches.

The output of the switch must be a constant positive tension when the input is negative and zero when the input is positive.

The resulting general computer schematic is given in Figure 2.

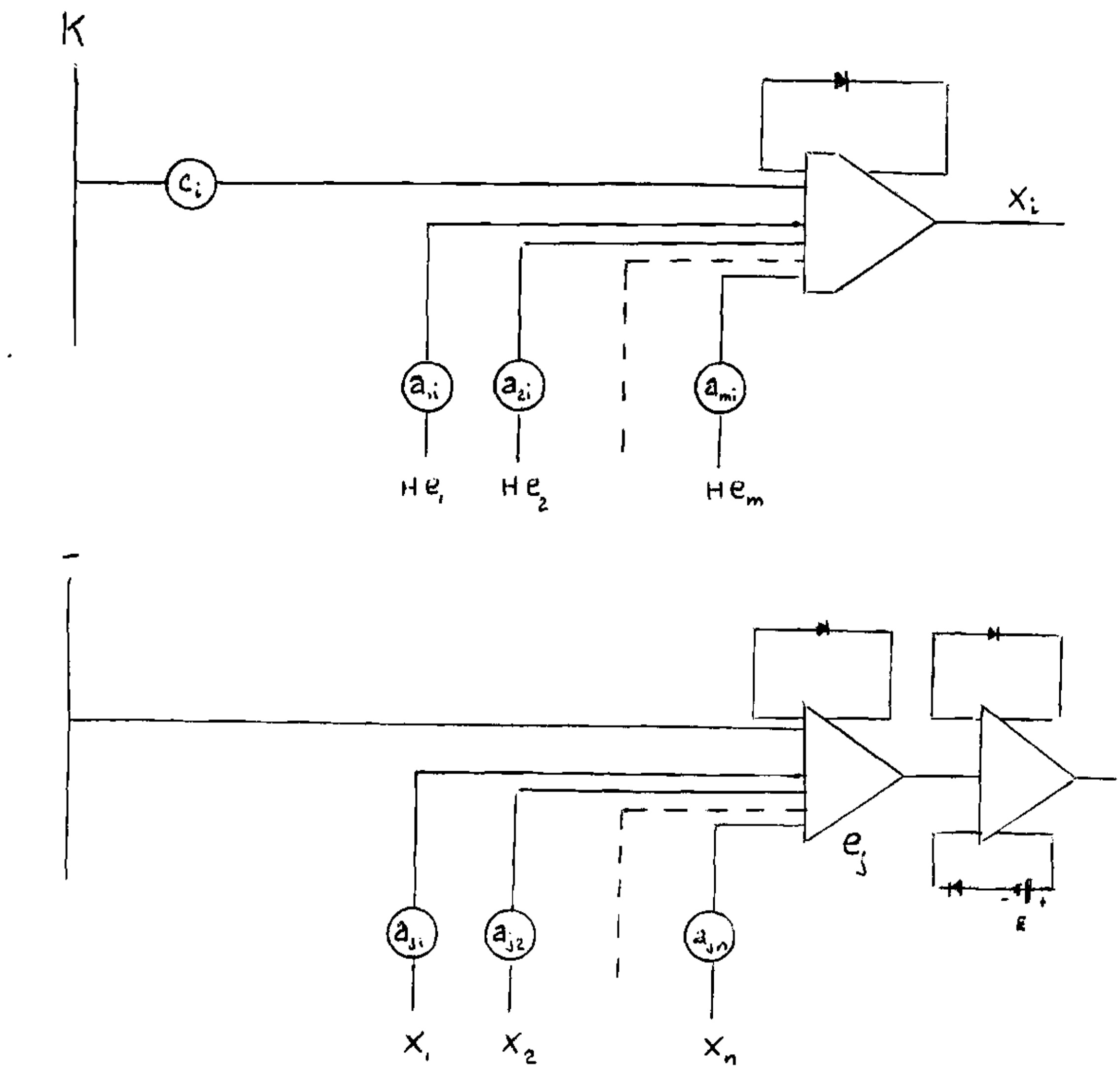


Figure 2. General Computer Schematic for Pyne's Method

# CHAPTER III

## JACKSON'S METHOD PERFORMANCES

### The Block Diagram

The motion of the objective point can be described without loss of generality by the equation

$$\frac{dr}{dt} = \text{grad } Z - K \sum_{j=1}^{m} d_j(A_j - R_j)\text{grad } A_j \qquad (3\text{-}1)$$

where $R_j$ is a constant,

$$A_j = \sum_{k=1}^{n} a_{jk}x_k$$

and

$$\text{grad } A_j = \sum_{k=1}^{n} \frac{\partial A_j}{\partial x_k} i_k = a_{j1}i_1 + a_{j2}i_2 + \dots + a_{jn}i_n$$

Let

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

$$\underline{\dot{x}} = \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \vdots \\ \frac{dx_m}{dt} \end{bmatrix}$$

$$c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

Let

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

where

$$b_{ij} = a_{ij}$$

if the *ith* constant is active, and

$$b_{ij} = 0$$

if it is not, and

$$B^T = \begin{vmatrix} b_{11} & b_{21} & \cdots & b_{m1} \\ b_{12} & b_{22} & \cdots & b_{m2} \\ \vdots & \vdots & & \vdots \\ b_{1m} & & \cdots & b_{mn} \end{vmatrix}$$

be the transpose of B.

If

$$M = \begin{vmatrix} R_1 \\ R_2 \\ \vdots \\ R_m \end{vmatrix}$$

is a column matrix of dimension m, equation (3-1) can be written

$$\dot{\underline{x}} = c - KB^T B\underline{x} + KB^T M \tag{3-2}$$

The matrix c represents the component of the velocity caused by grad z; the matrix $KB^T B\underline{x}$ the component originated from $K \sum_{j=1}^{m} d_j A_j$ grad $A_j$; the matrix $KB^T M$ the component originated from $K \sum_{j=1}^{m} d_j R_j$ grad $A_j$.

If the objective point is in the interior of the feasible region, all the elements of B are equal to zero. When only one constraint is active B is a row matrix; when p constraints are active, it is a matrix of dimensions $p \times n$. As the point moves, more and more constraints become active, and the number of rows of matrix B increases. The optimal point corresponds to the intersection of n hyperplanes, as many as the number of dimensions of the space. The objective point stops in its proximity, where n constraints are violated. Therefore, at the optimal point the matrix B is a square matrix of dimension n. This corresponds to the decomposition of the vector grad z along n linearily independent directions, which are normal to the active constraints.

The block diagram corresponding to equation (3-1) is given in Figure 3.

## Movement of the Objective Point When the First Constraint is Reached

If $A_j < R_j$ for $j \neq 1$ and $A_1 \geq R_1$, equation (3-1) becomes

$$
\begin{aligned}
\dot{x}_1 &= c_1 - Ka_{11}(a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n) + Ka_{11}R_1 \\
\dot{x}_2 &= c_2 - Ka_{12}(a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n) + Ka_{12}R_1 \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\dot{x}_n &= c_n - Ka_{1n}(a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n) + Ka_{1n}R_1
\end{aligned}
\qquad (3\text{-}3)
$$

For simplicity in the remaining part of this section, the constants $a_{ij}$ will be written $a_j$.

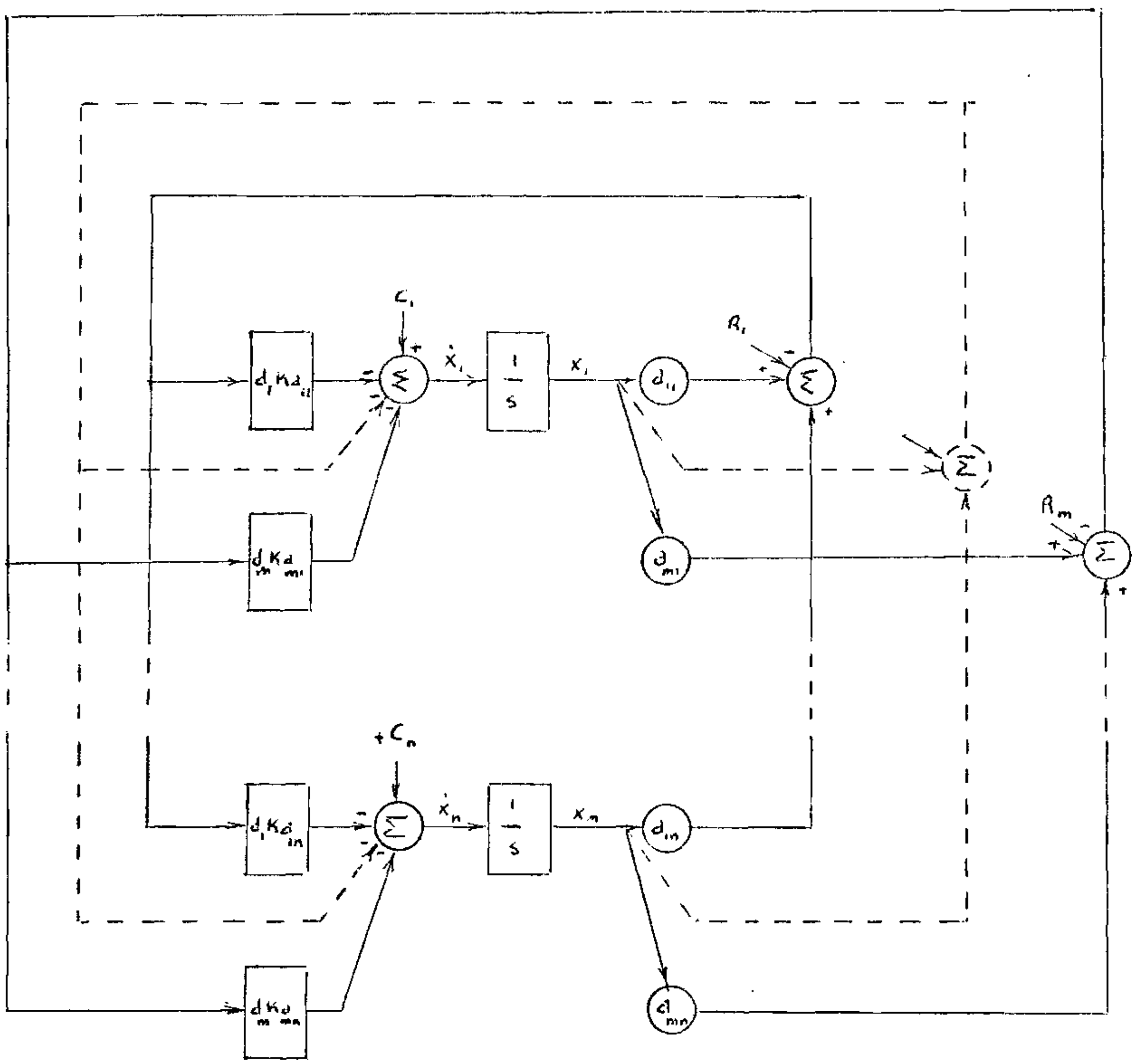


Figure 3. Jackson's Method Block Diagram

Let P, of coordinates $x_{01} = 0$, $x_{02} = 0$, ..., $x_{0n} = \frac{R_1}{a_n}$, be a point on the boundary, i.e.

$$x_{01}a_1 + x_{02}a_2 + \dots + x_{0n}a_n = R_1$$

This will be the position occupied by the objective point at time zero (initial condition). Equation (3-3) can be written

$$\dot{\underline{x}} = A\underline{x} + D \tag{3-4}$$

where

$$A = -\begin{vmatrix} Ka_1^2 & Ka_1a_2 & \cdot & Ka_1a_n \\ Ka_1a_2 & Ka_2^2 & \cdot & Ka_2a_n \\ \cdot & \cdot & \cdot & \cdot \\ Ka_1a_n & Ka_2a_n & \cdot & Ka_n^2 \end{vmatrix}$$

$$D = \begin{vmatrix} K_1R_1 + C_1 \\ Ka_2R_1 + C_2 \\ \vdots \\ Ka_nR_1 + C_n \end{vmatrix}$$

Taking the Laplace transform of equation (3-4) and solving for X:

$$X = |Is-A|^{-1}\underline{x}_0 + |Is-A|^{-1}\frac{D}{s}$$

where I is the identity matrix, and

$$\underline{X}_0 = \begin{vmatrix} 0 \\ 0 \\ \vdots \\ \frac{R_1}{a_n} \end{vmatrix}$$

The term $|Is-A|^{-1}\underline{X}_0$, which depends on the initial conditions, is called free response.

The term $|Is-A|^{-1} \frac{D}{s}$, independent of the initial conditions, is called forced response.

The matrix $|Is-A|^{-1}$ is given in Table 1. For sake of simplicity, let us assume, without loss of generality:

$$a_1^2 + a_2^2 + \ldots + a_n^2 = 1 \tag{3-5}$$

$$c_1a_1 + c_2a_2 + \ldots c_na_n = 1 \tag{3-6}$$

Equation (3-5) states that the vector $N_1$ has unit module. Equation (3-6) states that the projection of the gradient on the direction of $N_1$ has unit length.

The Laplace transform of the free response is

$$X_i = - \frac{a_iR_i}{s} + \frac{a_iR_1}{s + K} \qquad i \neq n$$

$$X_n = -\frac{a_n R_n}{s} + \frac{a_i R_1}{s+K} + \frac{R_1}{a_n s}$$

Table 1. $|Is-A|^{-1}$ Matrix in Jackson's Method

$$\begin{vmatrix} \dfrac{s + K\sum\limits_{i\neq 1} a_i^2}{s(s+K\sum\limits_{i=1}^{n} a_i^2)} & -\dfrac{Ka_1a_2}{s(s+K\sum\limits_{i=1}^{n} a_i^2)} & \cdot & -\dfrac{Ka_1a_n}{s(s+K\sum\limits_{i=1}^{n} a_i^2)} \\ -\dfrac{Ka_1a_2}{s(s+K\sum\limits_{i=1}^{n} a_i^2)} & \dfrac{s + K\sum\limits_{i\neq 2} a_i^2}{s(s+K\sum\limits_{i=1}^{n} a_1^2)} & \cdot & -\dfrac{Ka_1a_2}{s(s+K\sum\limits_{i=1}^{n} a_i^2)} \\ \cdot & \cdot & \cdot & \cdot \\ -\dfrac{Ka_1a_n}{s(s+K\ \ a_i^2)} & -\dfrac{Ka_2a_n}{s(s+K\sum\limits_{i=1}^{n} a_i^2)} & \cdot & \dfrac{s + K\sum\limits_{i\neq n} a_i^2}{s(s+K\sum\limits_{i=1}^{n} a_i^2)} \end{vmatrix}$$

The Laplace transform of the forced response is

$$X_i = \frac{1}{s^2}(c_i - a_i) + \frac{(1+KR_1)a_i}{sK} - \frac{1}{s+k}\,\frac{(1+RK)a_i}{K} \qquad 1 \le i \le n$$

The Laplace transform of the total solution is

$$X_i = \frac{1}{s^2}(c_i - a_i) + \frac{a_i}{Kr} - \frac{a_i}{K(s+K)} \qquad i \neq n$$

$$X_n = \frac{1}{s^2}(c_n - a_n) + \frac{a_n}{Ks} - \frac{a_n}{K(s+K)} + \frac{R_1}{a_n s}$$

and in the time domain

$$x_i = (c_i - a_i)t + \frac{a_i}{K} - \frac{q_i}{K} e^{-Kt} \qquad i \neq n$$

$$x_n = (c_n - a_n)b + \frac{a_n}{K} - \frac{a_n}{K} e^{-Kt} + \frac{R}{a_n}$$

Each of the coordinates of the objective point is given by the sum of

a. A term increasing linearily with time.

b. A constant term.

c. A decreasing exponential.

The term increasing linearily with time corresponds to a constant component of the velocity. The constant term is proportional to the distance of the objective point from the allowed region, and it corresponds to an error. This error is reached after a transitient, given by the exponential term. The constant component of the velocity is the projection of grad z on the boundary hyperplane. This can be proved considering that the components of this vector are the form $c_i - a_i$, i.e. it is obtained adding to grad z, of components $c_i$, a vector N of components $a_i$, and remembering that the projection of grad z in the direction of the unit vector N, which is normal to the boundary, is of length one.

It should be noted that the error and the constant of time of the transient are inversely proportional to the constant K. To make a little error, and to reach quickly the steady state, it is necessary to choose the constant K as big as possible.

Final Error Evaluation

The final position of the objective point can be found setting $\dot{\underline{x}} = 0$ in equation (3-2):

$$0 = c - KB^T B\underline{x} + KB^T M \tag{3-7}$$

It is near, but it does not coincide with the point whose coordinates maximize the objective function and satisfy the constraints; this causes an error.

When the coordinates of the objective point are sufficiently near to their final value, n constraints are active, and matrix B in equation (3-7) is a square matrix of dimension n, whose determinant will be called $|B|$.

Suppose the optimal point is the intersection of the first n constraints. With a proper translation of axes it is possible to move the origin of the coordinate axes to the optimal point. In this case, the constant in the equation of the first n constraints becomes zero, and equation (3-7) becomes

$$0 = c - KB^T B\underline{x}' \tag{3-8}$$

In the remaining part of this section, we shall make reference to those coordinates, and we'll write them without the prime notation.

The error in each of the coordinates is given by the values which satisfy the equation

$$B^T B\underline{x} = \frac{c}{K}$$

From Kramer's rule it is clear that the error will be inversely proportional to K, as it was said in the previous section, and $|B|^2$.

The last statement has a geometrical interpretation. The elements of B are the components of the vectors $N_j$ normal to the boundary hyperplane containing the optimal point. $|B|$ is the volume of an n dimension parallelepiped whose edges are those vectors. If the angle between the boundary hyperplanes tends to zero this volume decreases, and the error tends to infinity. Therefore, Jackson's method should not be used in those cases.

It is possible to reduce the error allowing K to change discontinually from zero to a large value. This can be accomplished by using switches to determine when the feedback loops are activated. The limit of this method consists in the fact that K cannot increase indefinitely, or the computer amplifiers are overloaded.

Introducing the constant H and K defined in Chapter II, the movement equation becomes:

$$\dot{\underline{x}} = H \text{ grad } z + K \sum_{i=1}^{m} d_i N_i (A_i - 1)$$

The final position of the objective point is reached through the opposite effect of H grad z and $K \sum_{i=1}^{m} d_i N_i (A_i - 1)$. Instead of increasing the value of K, a better method to reduce the error is to decrease the value of H. H is kept large in the allowed region and when few constraints are active, so that the objective point reaches quickly the neighborhood of the optimal point; then it is reduced. If the reduction is too great, the error caused by disturbances becomes excessively great. The reduction of H can be done manually, regulating the potentiometers.

# CHAPTER IV

## PYNE'S METHOD PERFORMANCES

### Block Diagram

A greater number of amplifiers are necessary to solve an optimization problem with Pyne's method than with Jackson's. That is because in the former every switching function requires an amplifier, while in the latter it can be implemented by adding a diode to the feedback loop of the existing adder.

The block diagram of the equation describing Pyne's method is given in Figure 4.

### Velocity of the Objective Point

Denoting with V the speed of P, the objective point when no constraints are in play, suppose that P has to move at an angle $\theta$ to its unconstrained direction.

It is easy to see that the effect of the boundaries will cause P to move with the speed $V \cos \theta$ in the new direction, because the component of the velocity originated by the constraints is normal to the constraints themselves. The total speed is then the time average of two velocities acting alternately. One is grad z, the other the sum of grad z and of the reaction of the constraints. Those two velocities have the same component along the direction of the total speed: $V \cos \theta$.

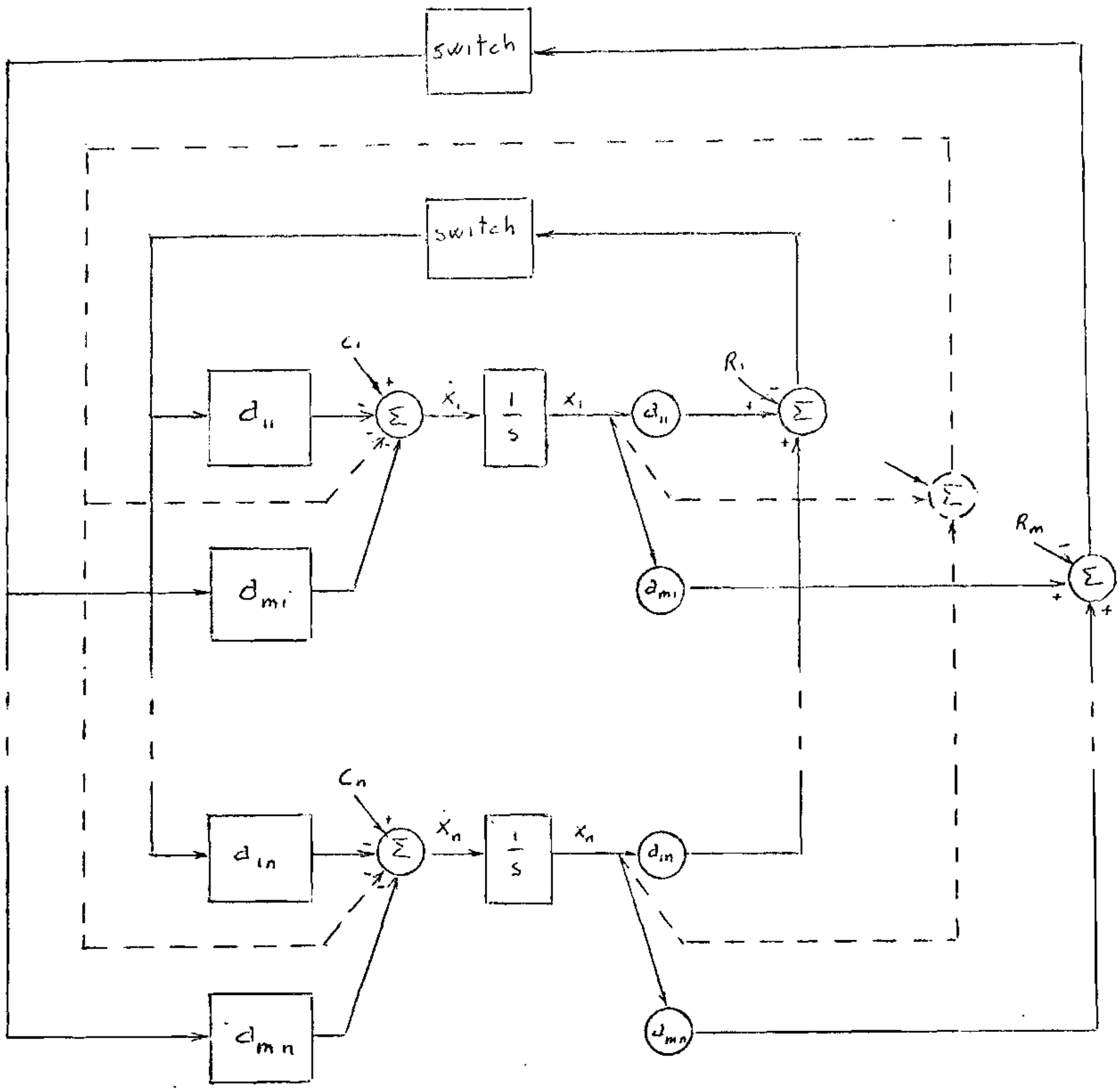


Figure 4. Pyne's Method Block Diagram

The component of the velocity in the direction of the gradient is therefore reduced of a factor $\cos^2\theta$. In the worst case P travels along the one-dimensional intersection of n-1 hyperplanes. We can obtain an average estimate of its velocity by finding the expected value of $V \cos^2\theta$ for a line selected at random in the n-space. The calculation is straightforward (2) and yields the expected value

$$\overline{V \cos^2\theta} = \frac{\int V \cos^2\theta \, dS}{\int dS} = \frac{V}{n},$$

the integration being performed over the surface of the unit sphere in n-space.

Although this number is small when n is large, few problems may be expected in which the solution would be slowly approached, because this case will occur only toward the end of the problem and, in any event V can be made quite large.

The movement of the objective point can be described by the equation:

$$\dot{P} = K \text{ grad } z + \sum_{j=1}^{m} d_j N_j$$

To prevent breakthrough into the restricted region $N_j$ must be greater than the component of K grad z normal to the boundary

$$-K \text{ grad } z \cdot N_j \leq N_j^2$$

Using equations (2-3) and (2-5), this determines an upper limit on K

$$K \leq \frac{\sum_{K=1}^{n} a_{jk}^{2}}{\sum_{K=1}^{n} a_{jk} c_k}$$

which must be satisfied for every j.

## Errors

The objective point does not stop in proximity of the optimal point, but goes on bouncing from allowed to not allowed region, and vice versa. The amplitude of this bouncing is independent of the size of the vectors K grad z and $\sum_{j=1}^{m} d_j N_j$, but depends on the sensibility of the computer circuits. In other words there is no analytic error; this is a clear advantage over Jackson's method in which, as it has been shown, the greatest part of the error is not caused by sensibility, but by the chosen movement equation.

# CHAPTER V

# USE OF SECOND ORDER MOVEMENT EQUATIONS

## The Movement Equation

One of the limitation of the two methods presented previously is that the velocity of the objective point can be very small, and consequently the time necessary to reach the optimal point can be exceedingly long.

The worst condition is when the objective point moves along a long edge, forming with the vector grad z, an angle very near to $\frac{\pi}{2}$. This drawback can be avoided using a second order differential equation, i.e. describing the n-dimensional space as a field of accelerations, and not of velocities. A second advantage of this approach is that it is possible to reduce the duration of the transients originated when a new constraint becomes active.

Let the variable y be subject to the equation:

$$\ddot{y} = f_n^2 x - f_n^2 y - 2bf_n \dot{y} \qquad (5\text{-}1)$$

where $f_n$ and b are constants.

The corresponding block diagram is shown in Figure 5. Assuming

$$0 \le b \le 1$$

the characteristic equation for equation (5-1) is

$$D^2 + 2bf_nD + f_n^2 = (D+bf_n-jf_n\sqrt{1-b^2})(D+bf_n+jf_n\sqrt{1-b^2}) = 0$$

hence the roots are

$$D_1 = -bf_n + jf_n\sqrt{1-b^2} = -a + jf_d$$

$$D_2 = -bf_n - jf_n\sqrt{1-b^2} = -a - jf_d$$

where $a = f_nb$ is called the dumping coefficient, and $f_d = f_n\sqrt{1-b^2}$ is called the dumped natural frequency; a is the inverse of the constant of time of the system.

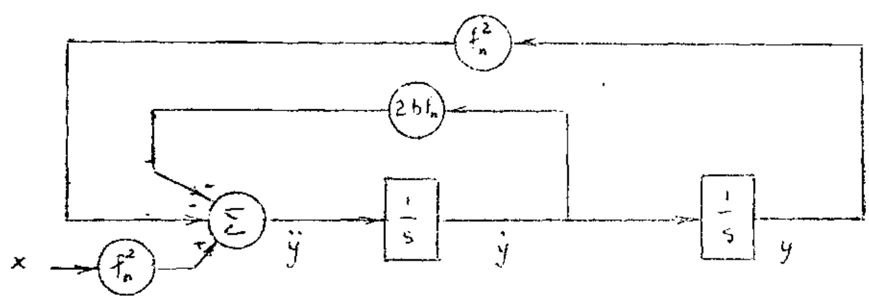


Figure 5. Second Order System

The unit step response, that is, the values that y assumes when x is a unit step, is given by

$$y(t) = -\frac{f_n}{f_d} e^{-at} \operatorname{sen}(f_d t+g) \tag{5-2}$$

where $g = \operatorname{arctg} \frac{f_d}{a}$ .

Figure 6 is a parametric representation of the unit step response. Note that the abscissa of this family of curves is the normalized time $f_n t$, and the parameter defining each curve is the dumping ratio b. Equation (5-1) suggests how to modify equation (2-7).

The movement of the objective point in the n space is described by the equation

$$\frac{d^2 r}{dt^2} = H \operatorname{grad} z - K_2 \frac{dM}{dt} - K_1 m$$

where

$$M = \sum_{i=1}^{m} d_i (A_i - R_i) \operatorname{grad} A_i$$

Equation (5-2) is equivalent to the set of n equations

$$\ddot{x}_1 = c_1 - K_2 \sum_{i=1}^{m} a_{i1} d_i \sum_{k=1}^{n} a_{ik} \dot{x}_k - K_1 \sum_{i=1}^{m} d_i a_{i1} \Big( \sum_{k=1}^{n} a_{ik} x_k - R_i \Big)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \tag{5-3}$$

$$\ddot{x}_2 = c_n - K_2 \sum_{i=1}^{m} a_{in} d_i \sum_{k=1}^{n} a_{ik} \dot{x}_k - K_1 \sum_{i=1}^{m} d_i a_{in} \Big( \sum_{k=1}^{n} a_{ik} x_k - R_i \Big)$$

and introducing the n auxiliary variables $y_1, y_2, \ldots, y_n$:

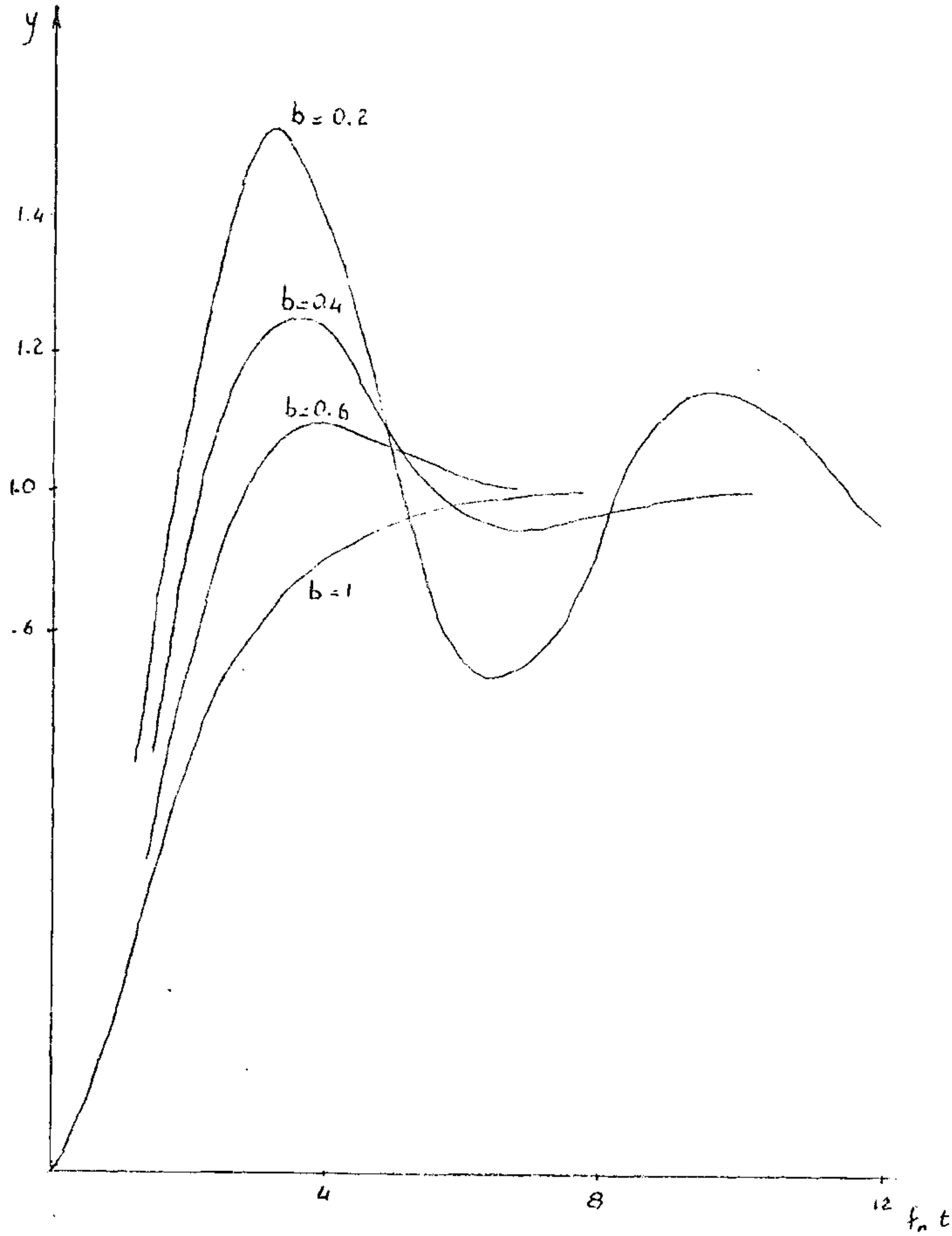


Figure 6. Second Order System Response

$$\dot{y}_1 = c_1 - K_2 \sum_{i=1}^{m} a_{i1} d_i \sum_{K=1}^{n} a_{ik} y_k - K_1 \sum_{i=1}^{m} d_i a_{i1} \left( \sum_{K=1}^{n} a_{ik} x_k - R_i \right)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \qquad (5\text{-}4)$$

$$\dot{y}_n = c_n - K_2 \sum_{i=1}^{m} a_{in} d_i \sum_{K=1}^{n} a_{ik} y_k - K_1 \sum_{i=1}^{m} d_i a_{in} \left( \sum_{k=1}^{n} a_i x_u - R_i \right)$$

$$\dot{x}_1 = y_1$$

$$\cdot \quad \cdot$$

$$\dot{x}_n = y_n$$

The corresponding block diagram, when only one constraint is active, is given in Figure 7.

## Movement of the Objective Point When One Constraint is Active

Let only the $j$*th* constraint be active. For sake of simplicity in the remaining part of this section we shall write the constants $A_{jk}$, $R_j$, $d_j$ without the subscript j, i.e. they will be written $A_k$, R, d.

At time zero the objective point is on the constraint, and has a velocity of components

$$x_{0i} = b_i$$

With a translation of axis it is possible to make the origin coincide with this initial position. Using these new coordinates, the

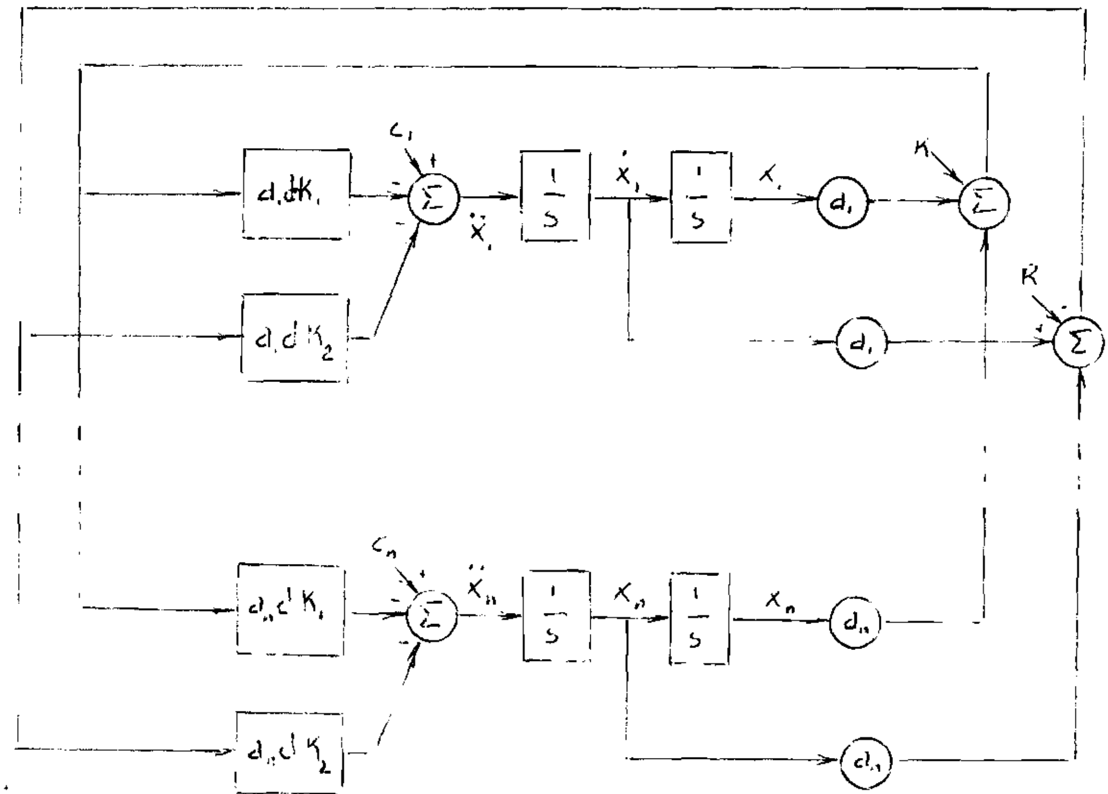

Figure 7. Second Order Movement Equation Block Diagram

initial position of the objective point is

$$x_{01} = 0$$

or it is: $R = 0$.

Equation (5-4) becomes

$$\dot{y}_1 = c_1 - K_2 a_1 d \sum_{K=1}^{n} a_k y_k - K_1 a_1 d\left(\sum_{K=1}^{n} a_k x_k - R\right)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\dot{y}_n = c_n - K_2 a_n d \sum_{K=1}^{n} a_k y_k - K_1 a_n d\left(\sum_{K=1}^{n} a_k x_k - R\right)$$

$$\dot{x}_1 = y$$

$$\cdots$$

$$\dot{x}_n = y_n$$

Proceeding in the same way as in Chapter II, under the assumptions

$$\sum_{i=1}^{n} a_1^2 = 1$$

$$\sum_{i=1}^{n} a_i c_1 = 1$$

it is possible to calculate the matrices $|A|$ and $|Is-A|^{-1}$. They are given in Tables 2 and 3.

Table 2. |A| Matrix When a Second Order Movement Equation is Used

$$
\begin{vmatrix}
-K_2a_1^2 & -K_2a_1a_2 & \cdot & -K_2a_na_1 & -K_1a_1^2 & -K_1a_2a_1 & \cdot & -K_1a_na_1 \\
-K_2a_1a_2 & -K_2a_2^2 & \cdot & -K_2a_na_2 & -K_1a_1a_2 & -K_1a_2^2 & \cdot & -K_1a_na_2 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
-K_2a_1a_n & -K_2a_1a_n & & -K_2a_n^2 & -K_1a_1a_n & -K_1a_2a_n & & -K_1a_n^2 \\
1 & 0 & & 0 & -s & 0 & & 0 \\
0 & 1 & & 0 & 0 & -s & & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & \cdot & 1 & 0 & 0 & \cdot & -s
\end{vmatrix}
$$

The forced response Laplace Transform is

$$X_i = \frac{a_i}{K_1}\frac{1}{s} + \frac{c_i - a_i}{s^3} - \frac{a_i}{K_1}\frac{s + K_2}{s^2 + K_2s + K_1}$$

The free response Laplace Transform is

$$X_i = \frac{b_i - a_i \sum_{h=1}^{n} a_hb_h}{s^2} + \frac{e_i \sum_{h=1}^{n} a_hb_h}{s^2 + K_2s + K_1}$$

The total response Laplace Transform is

Table 3. $|Is-A|^{-1}$ Matrix When a Second Order Movement Equation is Used

$$
\frac{1}{s^4+K_2s^3+K_1s^2}\left|\begin{array}{cccccc}
s^3+K_2(1-a_1^2)s^2+K_1(a-a_1^2)s & \cdot & -K_2s^2a_1a_n-K_1sa_1a_n & -K_1a_1^2s^2 & \cdot & -K_1a_1a_ns^2 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
-K_2s^2a_1a_n-K_1sa_1a_n & \cdot & s^3+K_2(1-a_n^2)s^2+K_1(1-a_1^2)s & -K_1a_1a_ns^2 & \cdot & -K_1a_n^2s^2 \\
s^2+K_2(1-a_1^2)s+K_1(1-a_1^2) & \cdot & -K_2sa_1a_n-K_1a_1a_n & s^3+K_2s^2+K_1(1-a_1^2)s & \cdot & -K_1a_1a_ns \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
-K_2sa_1a_n-K_1a_1a_n & \cdot & s^2+K_2(1-a_1^2)s+K_1(1-a_1^2) & -K_1a_1a_ns & \cdot & s^3+K_2s^2+K_1(1-a_1^2)s
\end{array}\right|
$$

$$X_i = \frac{a_i}{K_s} + \frac{c_i - a_i}{s^3} - \frac{a_i}{K_1} \frac{s + K_2}{s^2 + K_2 s + K_1} +$$

$$+ \frac{b_i - a_i \sum_{h=1}^{n} a_h b_h}{s^2} + \frac{a_i \sum_{h=1}^{n} a_h b_h}{s^2 + K_2 s + K_1}$$

The initial velocity of the forced movement is zero, but the objective point is subject to a constant acceleration of components $c_i - a_i$. Using the same considerations of Chapter III, it can be proved that the acceleration vector is given by the projection of grad z on the boundary plane. The effect of this acceleration is particularly useful when the constraint is almost normal to the gradient of the objective function. In this case, the objective point, instead of moving with a constant velocity grad z cos $\theta$, which is near to zero, increases its speed linearily with time. The term $a_k/K_1 s$ is proportional to the distance of the objective point from the constraint, and represents an error. This error is reached after a transient of the form

$$- \frac{a_i}{K_1} \frac{s + K_2}{s^2 + K_2 s + K_1}$$

With reference to equation (5-1) it is

$$K_1 = f_n^2$$

Transient duration is inversely proportional to $(K)^{1/2}$ (Figure 6), the error to $K_1$. In order to make a small final error and to have transients which fade in a short time, it is necessary to choose the constant $K_1$ as large as possible. To avoid big oscillations during the transient, and at the same time to reach in a short time the final value, 0.8 seems to be a proper value for b. Consequently, it is

$$K_2 = 1.6(K_1)^{1/2}$$

If a smaller value of $K_2$ is chosen the objective point is subject to dumped oscillations, that can give origin to spirals; if a greater value of K is chosen, the objective point does not oscillate but reaches slower the final position.

The inverse Laplace transform of the forced transient term is

$$\frac{a_1}{K_1} e^{-at} \cos \omega t + \frac{K_2}{(aK_1 - K_2^2)^{1/2}} \text{sen } \omega t \; e^{-at}$$

where

$$\omega^2 = K_1 - \frac{K_2^2}{4}$$

$$a = \frac{K_2}{2}$$

The free response is given by the sum of two terms. The first represents a constant velocity. This vector w is the projection of the initial velocity $V_0$ of components $b_i$ on the boundary hyperplane. To prove this statement, let us consider that the vector W is the sum of the vector $V_0$ and of a vector dV of components $a_1 \sum_{i=1}^{n} a_i b_i$. It is sufficient to show that dV is parallel to $N_j$, i.e. normal to the constraint, and of magnitude

$$\frac{N_j \cdot V_0}{|N_j|}$$

where $\cdot$ indicates the inner product; but this is immediate, remembering that $N_j$ is of unit module, and that

$$N_j \cdot V_0 = \sum_{i=1}^{n} a_i b_i$$

The second term represents a transient. The considerations that led to the choice of $K_1$ and $K_2$ hold also for this term. The inverse Laplace transform of the transient is of the form

$$\frac{2a_i \sum_{i=1}^{n} a_i b_i}{\sqrt{4K_1 - K_2^2}} \text{ sen } \omega t \; e^{-at}$$

It should be noted that the problem could not be solved with the use of only one feedback loop. If it is $K_2 = 0$, one obtains $a = 0$ and the objective point does not assume a fixed position, but goes on oscillating with frequency $\omega = (K_1)^{1/2}$ and amplitude $\frac{1}{K_1}$.

# CHAPTER VI

# PRIMAL AND DUAL PROBLEM SOLUTION

## Solution of a Compatible Set of Inequalities

There are some methods which allow to reduce some problems of linear programming to the solution of a system of inequalities. In this chapter the solution of linear programming problems will be considered only in the sense of the application of this method. The first section will consider the problem of solving a compatible system of inequalities irrespective of mathematical programming.

We consider a set of m compatible inequalities

$$A_i - R_i \leq 0 \qquad (6\text{-}1)$$

where $R_i$ is a constant.

The non-empty set of points satisfying Equation (6-1) is denoted by R. The problem is that of finding a point $X \epsilon R$. We reduce this problem to the one of finding the absolute minimum for the auxiliary function V(X), which is achieved in the points of R.

Let us introduce the function $H(A_i)$, which can be differentiated over $A_i$

$$H(A_i) = (A_i - R_i)^2 d(A_i - R_i)$$

where

$$d(A_i - R_i) = 0 \quad \text{for} \quad A_i \le R_i$$

$$d(A_i - R_i) = 1 \quad \text{for} \quad A_i > R_i$$

The function $H(A_i)$ is everywhere positive in the half space where the *ith* constraint is not satisfied, and identically equal to zero in the half space where it is.

Let us form the function

$$V(X) = \frac{1}{2} \sum_{i=1}^{m} (A_i - R_i)^2 d(A_i - R_i) \tag{6-2}$$

We have

$$\lim V(X) = +\infty \quad \text{for} \quad |X|^2 \text{ tending to infinity}$$

Outside R the convex function has no stationary points; therefore

$$|\text{grad } V(X)|^2 > 0 \qquad X \notin R$$

where grad $V(X)$ is the gradient of the function $V(X)$. Thus the inequalities problem (1) is reduced to that of finding the minimum points of the function $V(X)$.

To minimize the function $V(X)$, defined in section 2, one can use the gradient system of differential equations

$$\frac{dx_k}{dt} = -K \frac{\partial V(X)}{\partial x_K}$$

or, by (2)

$$\frac{dx_k}{dt} = - \sum_{i=1}^{m} \frac{\partial A_i}{\partial x_k} (A_i - R_i) d(A_i - R_i)$$

This equation has the same form as equation (2-7), in which grad z is set equal to zero, and can be interpreted in the same way.

## Solution of the Linear Programming Problem

Let us consider the primal and dual linear programming problems.

The primal problem:

Maximize

$$z(X) = \sum_{K=1}^{n} c_k x_k$$

over the set $R_1$ which is defined by the system of inequalities

$$\sum_{K=1}^{n} a_{ik} x_k - b_i \leq 0 \qquad 1 \leq i \leq m \tag{6-3}$$

$$-x_k \leq 0 \qquad 1 \leq k \leq n$$

The dual problem:

Minimize

$$W(Y) = \sum_{i=1}^{m} b_i y_i$$

over the set P, which is defined b the system of inequalities

$$-\sum_{i=1}^{m} a_{ik}y_i + c_k \le 0 \qquad 1 \le K \le n$$

$$- y_i \le 0 \qquad 1 \le i \le m \tag{6-4}$$

The dual problem is constructed according to the conditions of the primal problem, and it is known that

$$z(X) \le w(Y)$$

for $x \epsilon R$ and $y \epsilon P$. The equality holds for the optimal values of x and y, $x^*$ and $y^*$, that we are trying to determine.

The primal and dual problems can be reduced to one problem: find the vector $(x^*,y^*) = (x_1^*,x_2^*,\ldots,x_n^*,y_1^*,y_2^*,\ldots,y_n^*)$ which satisfies simultaneously the set of inequalities (3) and (4) and

$$-z(x) + w(y) = -\sum_{k=1}^{n} c_k x_k + \sum_{i=1}^{n} b_i y_i \le 0$$

The solution of the problems (3) and (4) can be reduced to the minimization of the function

$$v(x,y) = \frac{1}{2}\left\{\sum_{i=1}^{m}\left(\sum_{k=1}^{n} a_{ik}x_k - b_i\right)^2 d\left(\sum_{k=1}^{n} a_{ik}x_k - b_i\right) + \right.$$

$$+ \sum_{k=1}^{n}\left(-\sum_{i=1}^{m} a_{ik}y_i+c_k\right)^2 d\left(-\sum_{i=1}^{m} a_{ik}y_i+c_k\right) +$$

$$+ \left(\sum_{i=1}^{m} b_i y_i - \sum_{k=1}^{n} c_k x_k\right)^2 d\left(\sum_{i=1}^{m} b_i y_i - \sum_{k=1}^{n} c_k x_k\right) +$$

$$\left. + \sum_{k=1}^{n} x_k^2 d(-x_k) + \sum_{i=1}^{m} y_i^2 d(-y_i)\right\}$$

The gradient system looks like

$$\frac{dx_v}{dt} = -K\left\{\sum_{i=1}^{m} a_{iv}\left(\sum_{k=1}^{n} a_{ik}x_k-b_i\right)d\left(\sum_{k=1}^{n} a_{ik}x_k-b_i\right) +\right.$$

$$+ c_v\left(-\sum_{k=1}^{n} c_k x_k+\sum_{i=1}^{m} b_i y_i\right)d\left(-\sum_{k=1}^{n} c_k x_k+\sum_{i=1}^{m} b_i y_i\right) - \qquad (6\text{-}5)$$

$$\left. - x_v d(-x_v)\right\}$$

$$\frac{dy_j}{dt} = K\left\{\sum_{k=1}^{n} a_{jk}\left(-\sum_{i=1}^{m} a_{ik}y_i+c_k\right)d\left(-\sum_{i=1}^{m} a_{ik}y_i+c_k\right) +\right.$$

$$+ b_j\left\{\left(-\sum_{k=1}^{n} c_k x_k + \sum_{i=1}^{m} b_i y_i\right) d\left(-\sum_{k=1}^{n} c_k x_k + \sum_{i=1}^{m} b_i y_i\right) - y_i d(-y_i)\right\} \tag{6-6}$$

The system (6-6) can be easily implemented on an analog computer.

The computer schematic and block diagram are similar to those considered in Chapter II.

## Method Performances

In a number of applied problems, for example in digital optimization, in game and economical problems, the solution of both the primal and dual problems is required. In those cases this solution method is particularly useful.

The forcing function in the previous methods was represented by the gradient of the objective function. Here the movement of the objective point uniquely by the constraints. The only point where no constraints are active is the optimal point, which is the intersection of all non-redundant boundary hyperplanes. The solution is an exact one, as in Pyne's method, i.e. this procedure does not cause an error that cannot be eliminated because implicit in the motion equation. Unlike Pyne's method, the velocity by which the system attains the steady state can be increased by the proper use of a second order differential equation.

The price paid for the two advantages is a greater number of amplifiers: it is necessary to use

| | |
|---|---|
| n+m | integrators |
| n+m+1 | adders |
| 4P | inverters. |

# CHAPTER VII

## APPLICATIONS

In this chapter the problem:

$$\text{Maximize} \qquad z = x_1 + x_2$$

$$\text{subject to} \qquad 2x_1 + x_2 \leq 4$$

$$-x_1 + x_2 \leq 1,$$

whose solution is

$$x_1 = 1,\ x_2 = 2,\ z = 3,$$

will be solved using some of the methods previously presented, and the results will be compared. The extreme simplicity of this problem is caused by the limitations imposed by the small number of available amplifiers (14) and by the maximum number of inputs to each of them (three × 1, and two × 10). Ten potentiometers and six integrating networks were available, but these quantities were not critical in the determination of the size of the example problem.

## Jackson's Method

The computer schematic and the results are given in Figures 8, 9, 10, and 11.

The component of the objective point velocity caused by the constraint, that in the block diagram corresponds to the feedback variable, is the same in the two experiments reported. The term a, which represents the velocity component due to the objective function gradient, is varied. When it has a great value (a=1), the final position is reached in a shorter time, but the error is considerably greater than the case in which a is small (a=.5).

The procedure: start with a great value of a; when the velocity of the objective point is zero, reduce a, could seem the best. In practice, it gives unreliable results: the final value of the variable s depends greatly on the final value of a; if a becomes very small the error changes of sign, and the result that one obtains change from experiment to experiment.

## Pyne's Method

The computer schematic and the results are given in Figures 12 and 13.

The amplifiers in the feedback path, combined with the diodes, perform the switching function. Suppose the diode has an ideal transfer function, and the amplifier has a very great gain. If the objective point is in the allowed region, at the outputs of the summer and of the amplifier there is zero voltage. If it moves to the not allowed region at the output of the summer and the input of the amplifier there is a

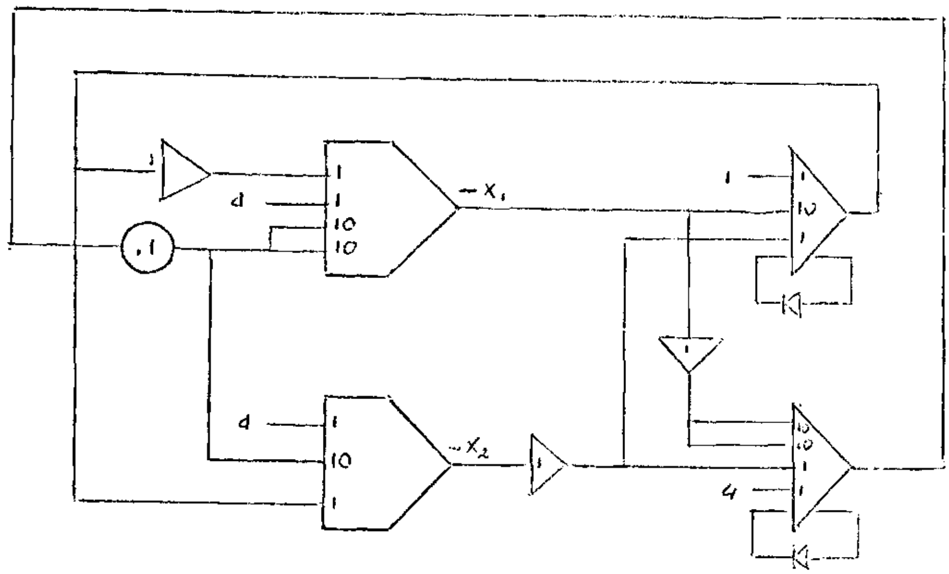


Figure 8. Jackson's Method Application Computer Schematic

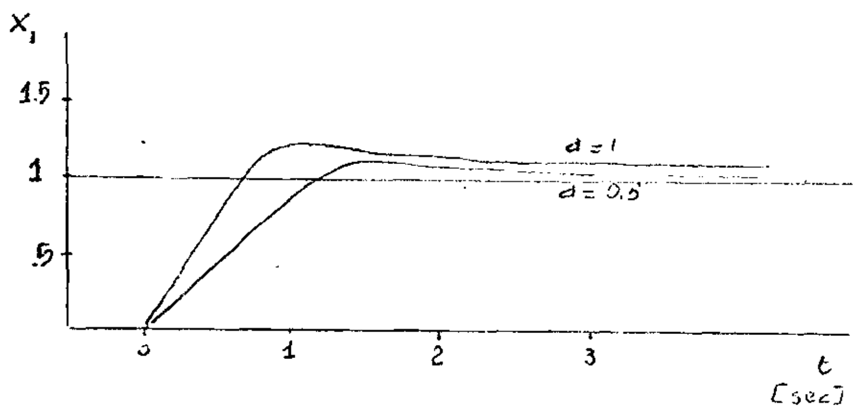


Figure 9. Jackson's Method Application Response—$x_1$

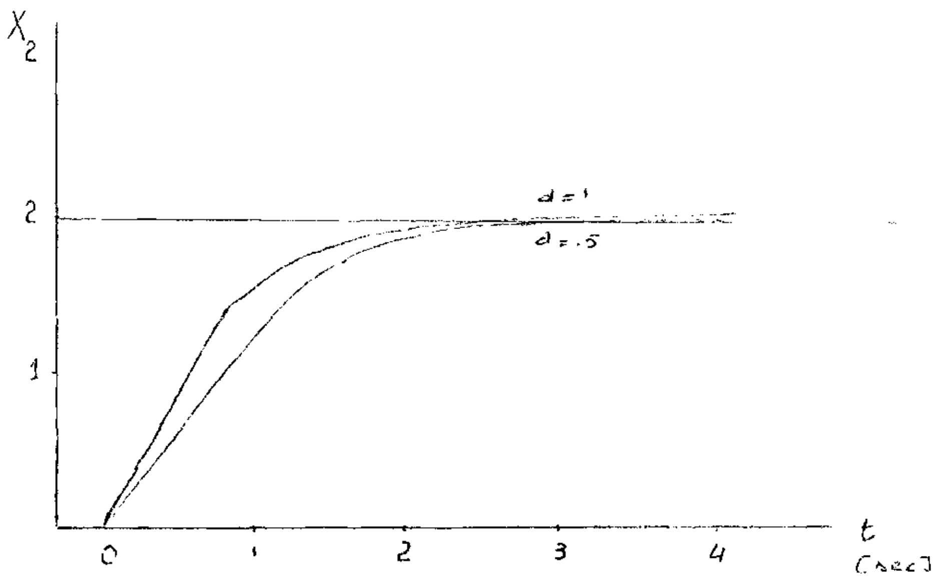


Figure 10. Jackson's Method Application Response—$x_2$

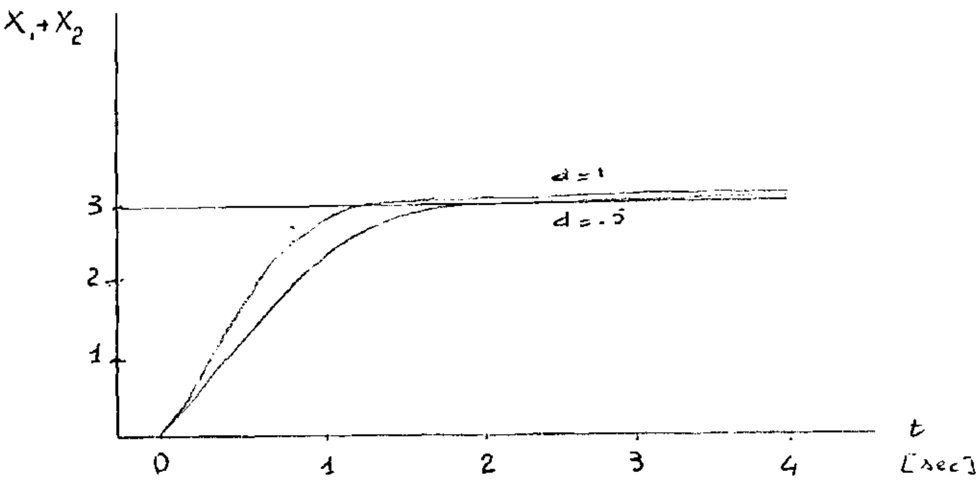


Figure 11. Jackson's Method Application Response—$z=x_1+x_2$

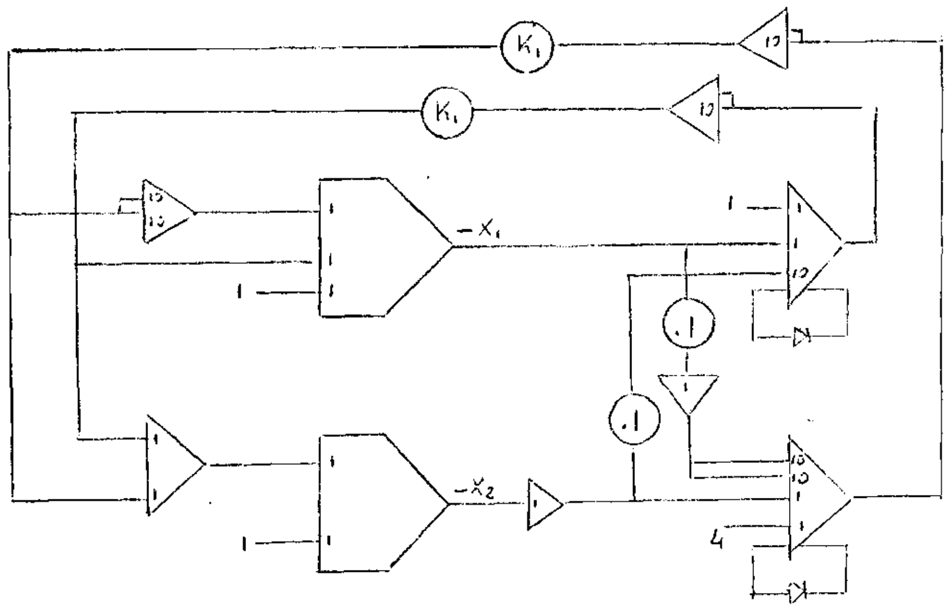


Figure 12. Pyne's Method Application Computer Schematic

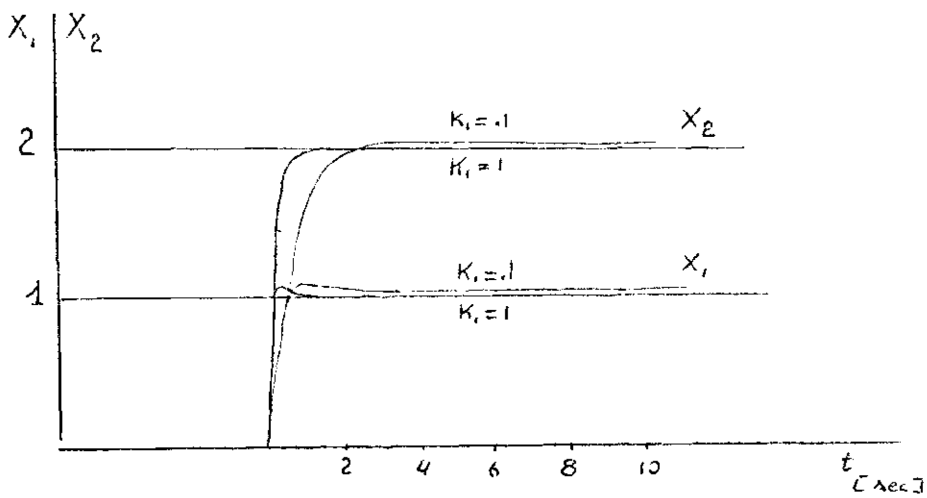


Figure 13. Pyne's Method Application Response—$x_1$ and $x_2$

voltage proportional to the distance from the boundary. At the output of the amplifier, remembering that the gain is very high, there is a constant voltage equal to the saturation tension. In reality the diode characteristic derivative has no discontinuities, and the gain of the amplifier is 20. The feedback tension is not a binary variable, but assumes all the values between the saturation voltages. In such a way, even if Pyne's method conceptually differs from Jackson's, in its practical application, it leads to the same computer set up, with the only difference of an increased feedback.

For $K_1$=1 the set up in Figure 12 is equivalent to the one in Figure 8, with a=1 and the feedback increased 20 times. For $K_1$=.1 the feedback is only twice as big. In this second case there is not a significant difference in the response of the two methods. In the first case there is no significant error, and the time required to obtain the solution is considerably shorter than with Jackson's method.

This result can be obtained because a large value of the constant a corresponds to a great velocity, and at the same time the increase in the feedback reduces the error.

## Second Order Equation Method

The computer schematic is given in Figure 14. The constant a is proportional to the component of the objective point acceleration given by the gradient of the objective function. If a=0 (Figures 15 and 16), while in the feasible region, the velocity should be constant and equal to $\dot{x}_{10}$, $\dot{x}_{20}$. In practice, as the switching function is not perfect, there is a small positive feedback, and the velocity increases slightly.

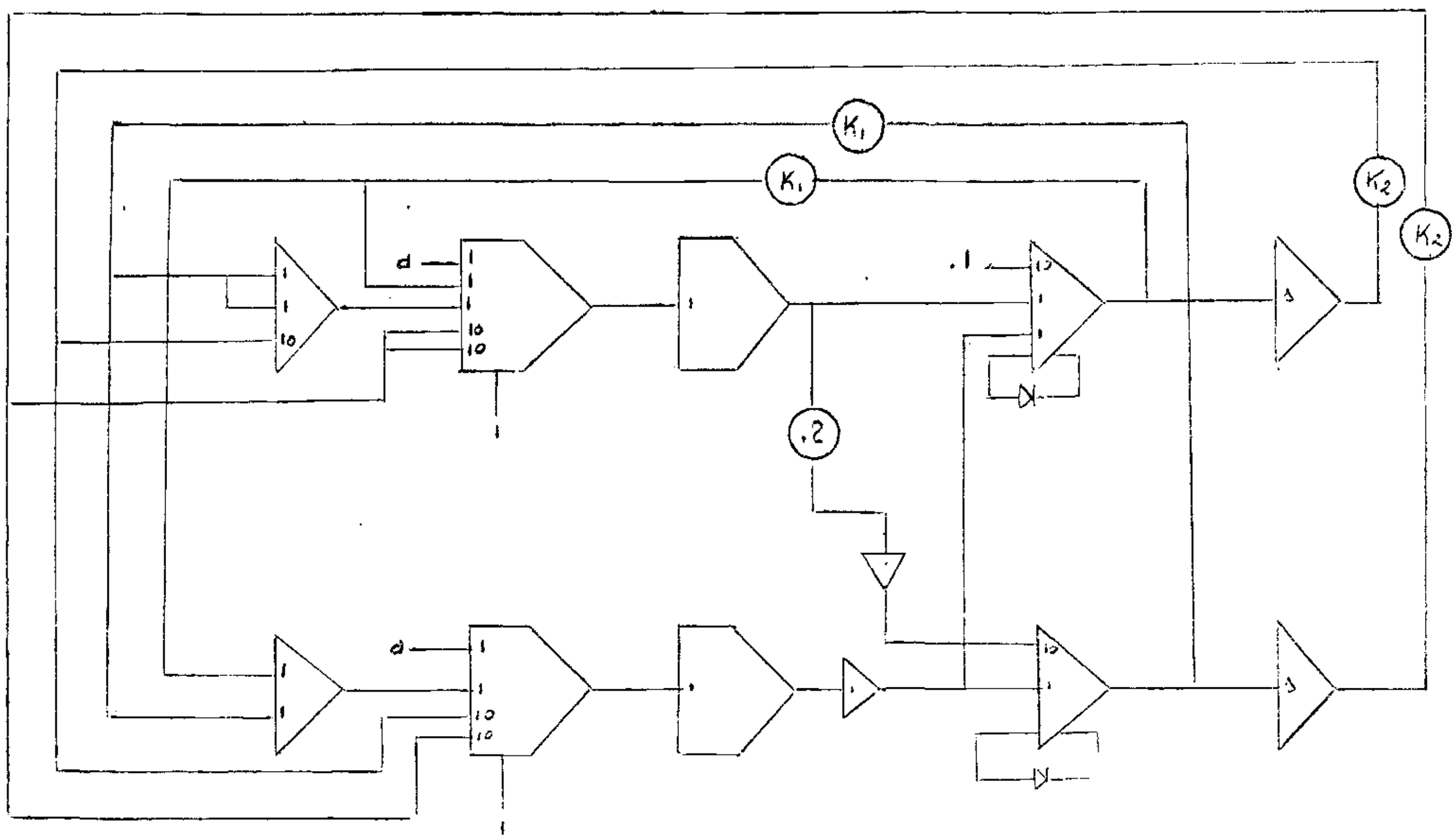


Figure 14. Second Order Equation Method Computer Schematic

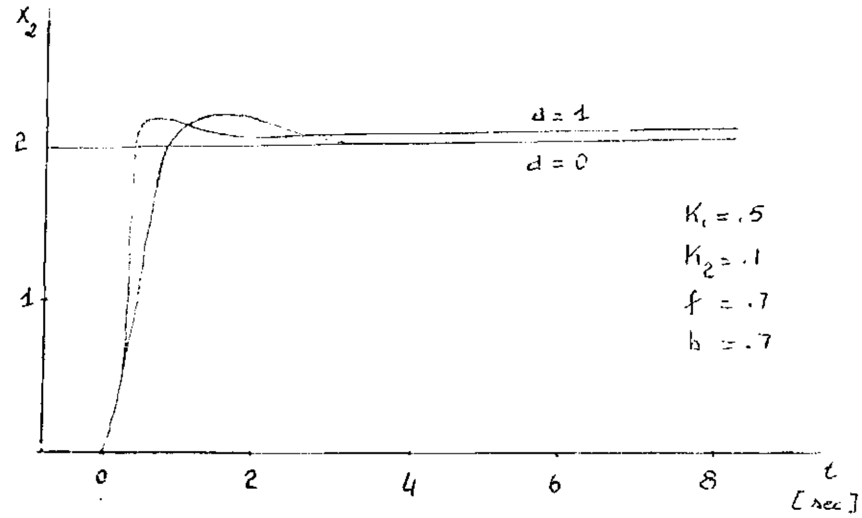


Figure 15. Second Order Equation Method Response: $x_2$

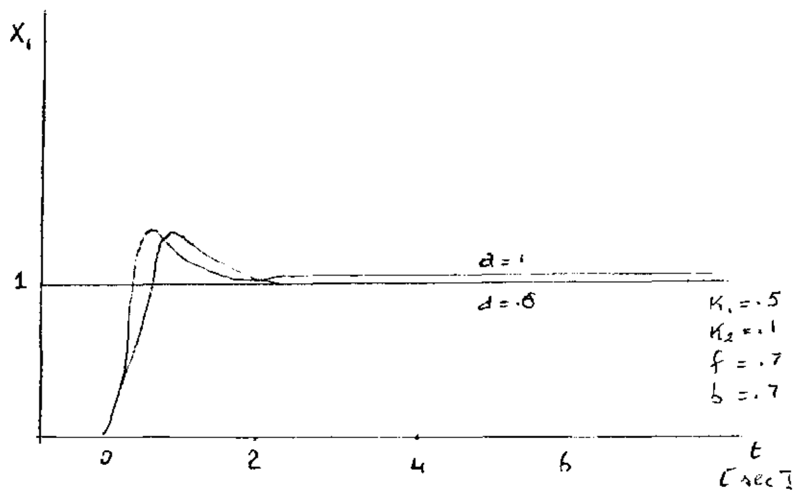


Figure 16. Second Order Equation Method Response: $x_1$

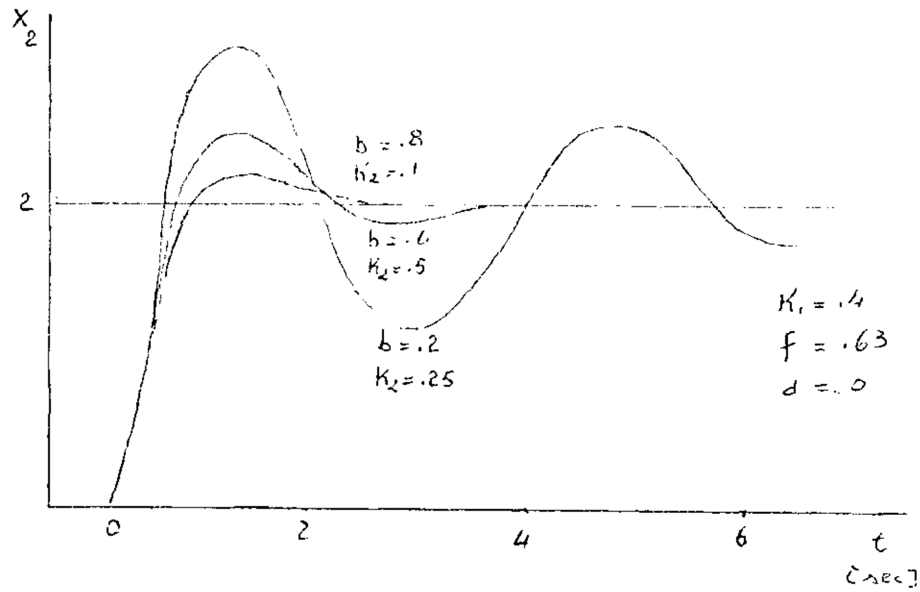


Figure 17. Second Order Equation Method Response: $x_2$

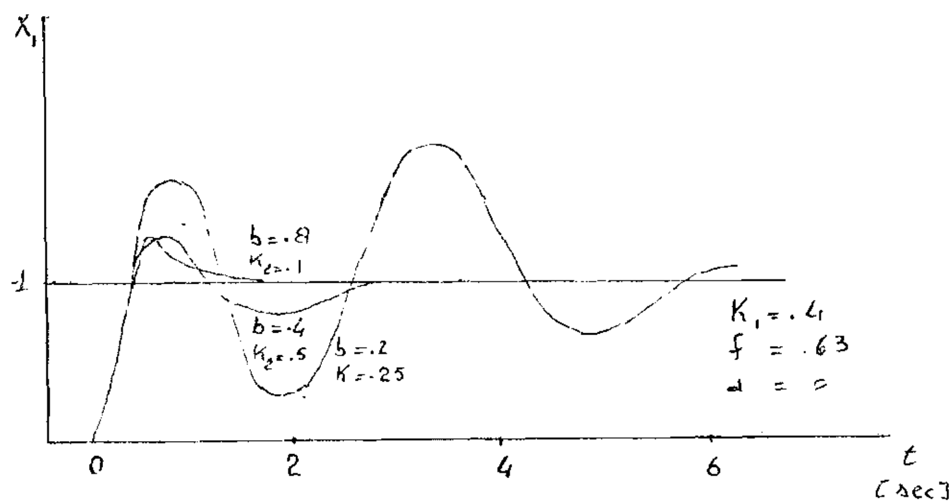


Figure 18. Second Order Equation Method Response: $x_1$

The error is zero, but to obtain the solution a longer time is required than in the case in which a greater value of a is used (a=1).

In Figures 17 and 18 are shown trajectories for various values of b. For b=.8 the results of this method are similar to those of Pyne's. It should be noted that the feedback in the latter was 40 times greater. It has not been possible, for lack of amplifiers, to increase the feedback in the former; probably this would lead to much better performances.

# CHAPTER VIII

# CONCLUSIONS

## Results

The method of solution of the programming problem on analog computer requires only 10 to 20 seconds for solution. This time is approximately independent of the size of the problem, and depends only on the required precision. This method is straightforward and requires no manipulation of the basic equation other than the introduction of magnitude scale factors. Once the problem is set up on the analog machine the solution time is on the order of a few seconds. This is in contrast to the relatively long time required for solution of such problems on a digital computer.

It is clear, of course, that the solution of such a problem requires a considerable amount of equipment. For instance, a problem whose basic matrix is 25×25 would require using Jackson's Method from 75 to 88 amplifiers, depending upon the number of negative coefficients in the basic matrix. This would require access to a medium-sized analog computer installation. It should be emphasized that the solution time of the problem, although it requires as many as 88 amplifiers, would be on the order of 10 seconds, as opposed to minutes in the case of a medium-sized digital computer. Once the problem is set up on analog computers, many variations of the problem can be run in a matter of minutes. This opens up a large field in the solution of programming problems. One may explore the sensitivity of the problem as a function of the parameters of the system and/or the effect on the system of predicted changes in the future completely at the will of operator. For example:

1. To move an edge parallel to itself, change the constant voltage applied to the corresponding summer. If necessary,change the corresponding resistance to prevent saturation of the amplifier.

2. To minimize instead of maximize an objective function, change the polarity of the constant voltage applied to integrators.

3. To interchange the restricted and allowed region for any edge: reverse input and output terminals of the diode.

4. To vary the slope of an edge, change the potentiometers corresponding to its equation coefficients.

5. If the integrating capacitors are all started from a discharged condition, the objective point will start its motion from the origin. It can be desirable to start from another initial condition; this can be done by introducing appropriate initial voltages on the integrating capacitors.

The methods outlined above for solution on an electronic analog computer have a very definite advantage over digital calculation, not only in speed, but also by virtue of the fact that the constraint and objective functions need not be linear in the region of interest. Most programming problems that are being solved today on digital machines are approximations of the actual situation, since the problem invariably must be linearized. This is a very decided advantage of analog computation.

## Limitations

Not all nonlinear problems, however, can be solved by analog methods. The requirements that must be satisfied, for obvious reasons, are:

1. The allowed region must be convex.

2. The objective function is unimodal.

3. Constraints and objective function are approximated by polynomials or by those irrational functions, such as square root, for which the function generator is available.

In the nonlinear case, using Jackson's method, the velocity vector is always defined by

$$\frac{dr}{dt} = H \text{ grad } z - K \sum_{j=1}^{m} d_j(A_j - R_j)\text{grad } A_i$$

where grad z is no more a constant, and $A_j$ is not a linear function.

In linear programming problems, as there is only one maximum in the solution space, the solution will be obtained on the first trial, starting from any initial point. Nonlinear problems, on the other hand, present two major difficulties. The first is that of describing the nonlinearities mathematically and being able to take these nonlinear functions into account during the calculations. This offers no basic difficulty to analog calculation because a nonlinear function is treated in the same manner as a linear function. The presence of nonlinear functions does, however, require more elaborate equipment.

The second difficulty is the fact that non-unimodal programming problems may have a number of relative or local maximum values as opposed to linear programming problems which always have one and only one maximum value of the objective function in the solution space.

Using any of the methods outlined above, any particular set of initial conditions will always produce the same solution, which may or may not be the absolute maximum. The obvious solution to this problem is to scan the solution space by using different sets of initial conditions for each trial. This method is not particularly efficient, and does not provide the assurance that all relative maxima have been found, even after a large number of trials. However, a very large number of trials can be carried out in a relatively short time, resulting in a good probability of finding the correct solution.

Another, and perhaps more serious limitation to the analog solution to mathematical programming problems is the time required for the computer set up. The analog computer set up is the equivalent to the programming of a digital computer. A program can be used for many problems, only changing the data characteristic to each of them and thus avoiding a great amount of work. In a similar way, the set up time in the linear case can be substantially reduced using a special problem programming board. In other words, the entire computer would be internally wired and its program board would then be a set of knobs, arranged and worked as in the final array. The set up time can be further reduced because, since each resistance among the ones corresponding to the coefficients $a_{ij}$ enters the problem twice, these could be combined so that both are set in a single operation. The special purpose analog computer can greatly reduce the set up time, but has two disadvantages over the general purpose one: the first is the lack of flexibility, i.e. it cannot be used for other problems or simulation studies; the second is the greater number of amplifiers that are required to solve a problem of given size. This fact could prevent its use in any reasonably sized problem. Consider, for example, a special problem computer wired according to Jackson's method, whose computer schematic is given in Figure 19.

Depending on the problem, every coefficient can be positive or negative. As a consequence, all connections must be wired twice: once directly, and once inserting an inverter. Depending on the sign of the coefficient in the problem under consideration, one of the connections is disconnected turning to zero the corresponding potentiometer.

Let m be the number of constraints, n the number of variables, z the maximum number of inputs to each amplifier. Let a be the minimum integer such that

$$2m \leq a(z-1) + 1$$

on b the minimum integer such that

$$2n \leq b(z-1) + 1$$

The total number of potentiometers required is

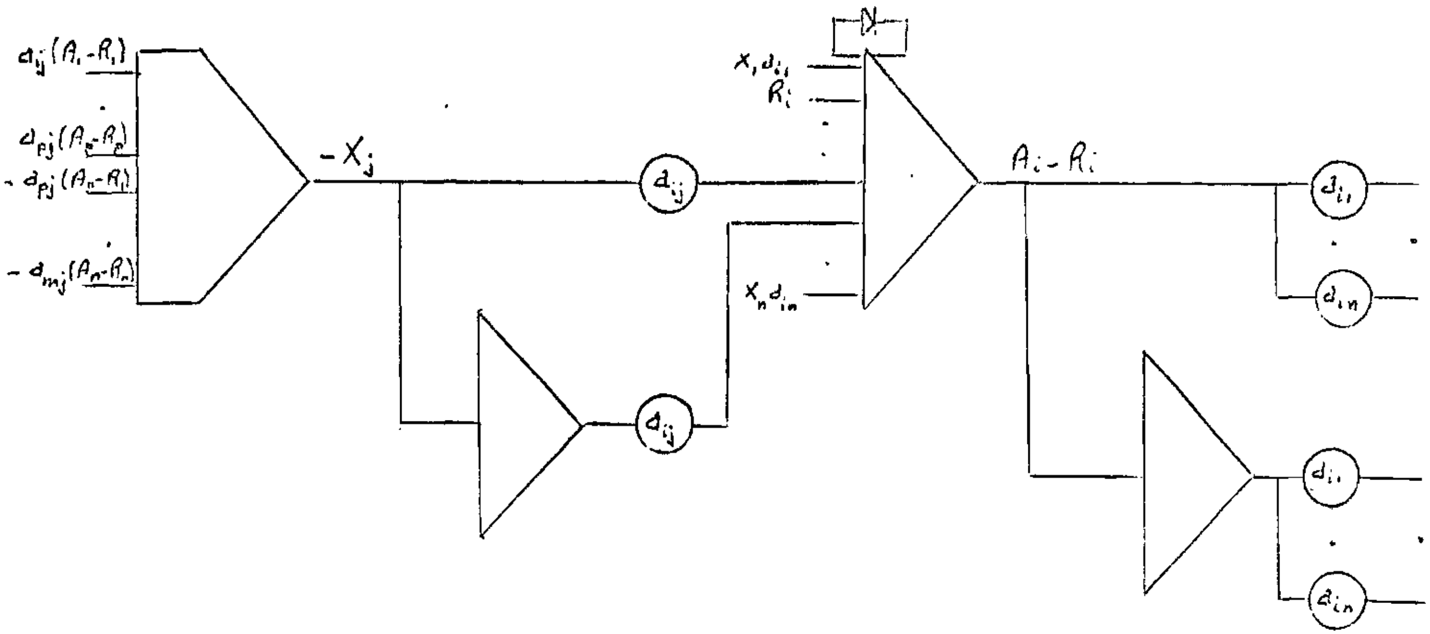


Figure 19. Jackson's Method Special Purpose Computer Schematic

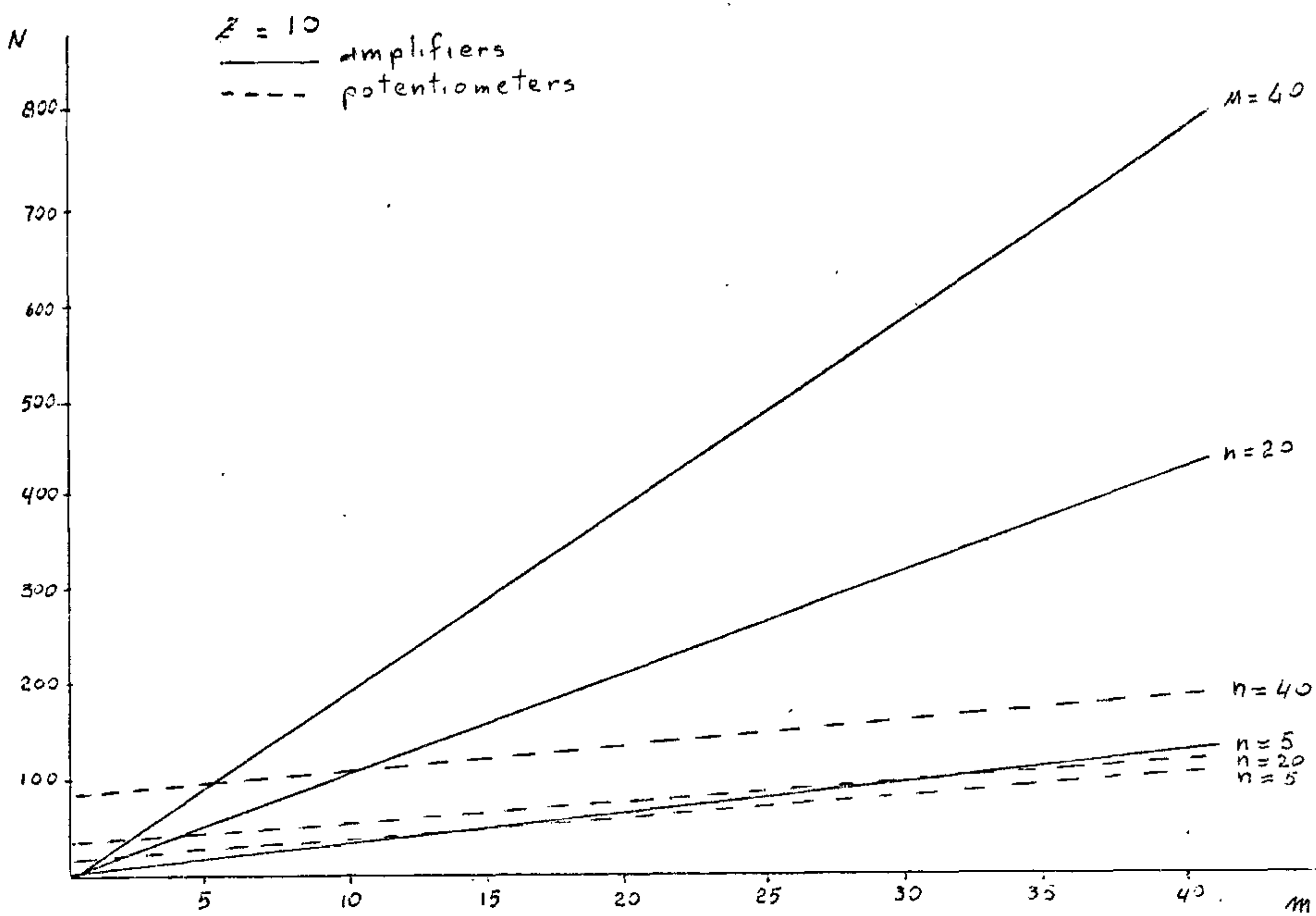


Figure 20. Number of Amplifiers and Potentiometers in Jackson's Method Special Purpose Computer

$$W = 2(n+m)$$

The total number of amplifiers required is

$$N = (a+1)n + (b+1)m$$

The values of N as a function of m, with n as a parameter, are given in Figure 10.

It is clear that N can reach very high values for small n and m. Probably this is one of the reasons analog solution methods to mathematical programming problems have received so little attention. It should be remembered, however, that it could be worth paying this price for the advantages offered in sensitivity analysis and nonlinear programming.

## Recommendations for Future Investigation

Two problems have not been considered in this thesis and should be studied:

1. Discussion of alternate methods of taking constraints into account. For example, the objective function could be modified in the following way:

$$\text{Max } z' = z - H \sum_{j=1}^{m} \frac{1}{A_j - R_j}$$

where H is a constant, and the solution could be found as the limit of Z' for H tending to zero.

2. Application of second order movement equation to the simultaneous solution of primal and dual problems.

It is also advisable to make an in-depth investigation of the nonlinear case; errors, velocity, and number of components should be determined; the possibility of sensitivity analysis should be examined.

# BIBLIOGRAPHY

1. A. S. Jackson, "Analog Computers Applied to Operations Research Research Problems," *ORSA Fifth Annual Meeting*, 10 May, 1957.

2. I. B. Pyne, "Linear Programming on an Analog Computer," *AIEE Trans.*, Part 1, Communications and Electronics, 1956.

3. R. P. Isaacs, *Differential Games*, New York, Wiley, 1965.

4. O. I. Elgert, *Control System Theory*, McGraw Hill Book Company, 1967.

5. J. J. DiStefano III, *Feedback and Control Systems*, McGraw Hill Book Company, 1967.

6. F. S. Hillier, *Introduction to Operations Research*, Holden-Day Inc., 1967.

7. W. A. Adock, "Automatic Simultaneous Equation Computer and Its Use in Solving Secular Equations," *Review of Scientific Instrumentation*, V. 19, 1948.

8. D. P. Campbell, "Dynamic Behavior of Linear Production Systems," *Mechanical Engineering*, April, 1953.

9. W. R. Fair, "Analogue Computations of Business Decisions," *Operations Research* I, 1963.

10. R. P. Rich, "Simulation as an Aid in Model Building," *Operations Research* 3, February, 1955.

11. C. J. Thomas and W. L. Deemer, "The Role of Operational Gaming in Operations Research," *Operations Research* 5, February, 1957.

12. G. K. Johnson and I. M. Turner, "Use of Transfer Functions for Comparing Planning," *Operations Research*, Vol. 4, No. 6, December, 1966.